UNITED STATES PATENT APPLICATION

of

Lih-Jyh Weng

for

VARIABLE REDUNDANCY CYCLIC CODE ENCODERS

VARIABLE REDUNDANCY CYCLIC CODE ENCODERS

BACKGROUND OF THE INVENTION

Field of the Invention

The invention relates generally to error correction systems and, more particularly,
5   to systems that use selected numbers of redundancy symbols.

Background Information

Data stored on magnetic media, such as magnetic disks, are typically stored in
encoded form, so that errors in the stored data can possibly be corrected. The errors may
occur, for example, because of inter-symbol interference, a defect in the disk, or noise. If
10   more errors are likely, and the system is thus required to correct greater numbers of
errors, more error correction information in the form of redundancy symbols is required
to be stored on the disk. Some of the trade-offs for including increased error correction
information on the disk are the extra time associated with encoding and decoding the data
and the complexity of the encoding and decoding systems.

15   Encoders for encoding data in accordance with cyclic error correction codes
("ECCs") to produce the redundancy, or ECC, symbols are well known.
An encoder that encodes the data using a given ECC manipulates the data symbols in
accordance with the corresponding generator polynomial. A conventional encoder that
encodes data in accordance with an ECC that produces "j" ECC symbols includes j
20   Galois Field multipliers. The Galois Field multipliers are associated, respectively, with
the j coefficients of the corresponding generator polynomial, g(x). The encoder also
includes j adders that combine the products associated with a given data symbol with the
results of the encoding of the previous data symbol, and j registers that hold the sums

1

produced by the adders. At the end of the encoding, the j registers contain the j ECC symbols.

Alternatively, the data may be encoded to produce the j ECC symbols using a circuit that includes, in addition to the j Galois Field multipliers, registers and adders, a chain of j feedback adders, as discussed in an article by Gerhard Fettweis and Martin Hassner entitled *A Combined Reed-Solomon Encoder And Syndrome Generator With Small Hardware Complexity*, published by IEEE in 1992. Such a circuit, referred to as a Fettweis-Hassner encoder/syndrome combination circuit, is used in systems in which the same Galois Field multipliers are used for both encoding and syndrome generation. The Fettweis-Hassner encoder/syndrome computation circuit, with its j Galois Field multipliers similarly produces j ECC symbols. However, as discussed in the article, the encoding is associated with a latency that is attributable to the delay through the chain of j feedback adders.

A conventional data processing system uses a selected encoder to encode n-j data symbols in accordance with the generator polynomial g(x) and produce the j ECC symbols that are to be combined with the data into n-symbol ECC code words. When a given data code word is later retrieved from the disk, the system uses the j ECC symbols to correct up to j/2 errors in the code word.

For certain media, however, ECCs that are capable of correcting more or fewer errors may be used to protect the data. For example, the media manufacturing process may produce media of varying quality. The media that meets the most stringent quality standards may require less error protection, and thus, require fewer ECC symbols. In contrast, the media that does not meet the same quality standards requires greater error protection and thus more ECC symbols.

Known prior systems use complex and/or time consuming encoding/decoding systems to produce the code words with different numbers of ECC symbols. One such system includes a plurality of encoders, each associated with a different generator polynomial. A user then selects which encoder to use to produce the required number of ECC symbols. The system may instead use a complicated encoding/decoding scheme in

2

which the user essentially deletes selected ECC symbols after encoding. This system uses a single encoder. However, when the data and the retained ECC symbols are later decoded, the system must use a relatively complex and time consuming error-and-erasure decoding scheme to perform error-only decoding.

5   If the Fettweis-Hassner encoder/syndrome combination circuit is used, the user may disable certain stages of the circuit, to select corresponding numbers of ECC symbols. The same stages are then used for syndrome generation during decoding. Such a system works well for ECCs that include relatively small numbers of ECC symbols, however, the delay associated with the feedback adder chain effectively prohibits the use

10   of the circuit for codes with relatively large numbers of ECC symbols.

Accordingly, what is needed is a less complex and/or time consuming encoding and decoding system that encodes the data to produce a desired number of ECC symbols, and decodes the data and ECC symbols using error-only decoding.

SUMMARY OF THE INVENTION

15   An encoder encodes the data in accordance with a selected number of factors of a cyclic code generator polynomial $g(x) = g_1(x) * g_2(x) * \ldots * g_f(x)$ to produce data code words that include a corresponding number of ECC symbols. The code words are thus produced by encoding the data in accordance with one of the polynomials:

$$p_1(x) = g(x) = g_1(x) * g_2(x) * \ldots * g_f(x)$$

20     $$p_2(x) = g_2(x) * g_3(x) * \ldots * g_f(x)$$

$$\ldots$$

$$p_f(x) = g_f(x)$$

$$p_{f+1}(x) = g_1(x) * g_2(x) g_3(x) * \ldots * g_{f-1}(x)$$

$$\ldots$$

25     $$p_m = g_1(x)$$

3

The encoder has multiple stages, and at least a first stage of the encoder corresponds to encoding by a factor $g_i(x)$ that is associated with a relatively large number of ECC symbols. The first stage is constructed to encode the data without significant delay, while other stages may be associated with longer delays but allow a greater selection of

5 associated numbers of ECC symbols, as discussed below.

The system uses the same circuitry to decode the code word in accordance with the particular polynomial selected to encode the data. Further, the system produces error syndromes and performs error location and correction operations in a conventional manner.

10 Briefly, the system produces a code word with a particular number of ECC symbols using, for example, a selected polynomial $p_t(x) = g_1(x)*g_2(x)*...*g_t(x)$ by first encoding the data $d(x)$ in a first stage using the factor $g_1(x)$ to produce

$$d(x)*x^s = q_1(x)g_1(x) + r_1(x),$$

where $q_1(x)$ is a quotient and $r_1(x)$ is a remainder and $g_1(x)$ has degree s. In a next stage

15 the encoder encodes $q_1(x)$ using the next factor of the selected generator polynomial, in the example, $g_2(x)$, to produce

$$q_1(x) = q_2(x)g_2(x) + r_2(x)$$

and so forth, until the remainders associated with all of the factors of the selected generator polynomial have been produced. The system then manipulates the remainders

20 associated with the factors to produce a remainder $r_t(x)$ that is associated with the selected polynomial, $p_t(x)$, and uses a cyclically shifted version of the remainder $r_t(x)$ as the code word ECC symbols.

As discussed in more detail below, the same circuitry is used to both produce and manipulate the remainders.

BRIEF DESCRIPTION OF THE DRAWINGS

The invention description below refers to the accompanying drawings, of which:

Fig. 1 is a functional block diagram of an encoder constructed in accordance with the invention;

Fig. 2 is a functional block diagram of a decoder for uses with the encoder of Fig. 1;

Fig. 3 is a functional block diagram of an alternative encoder; and

Fig. 4 is a functional block diagram of a decoder for use with the encoder of Fig. 3.

DETAILED DESCRIPTION OF AN ILLUSTRATIVE EMBODIMENT

The discussion below is separated into sections, a first section discusses the mathematical theory, a second discusses one embodiment of the encoder and a third section discusses another embodiment of the encoder. All mathematical operations are Galois Field operations.

A. The Theory

The k data symbols to be encoded may be represented as the coefficients of a degree k-1 polynomial d(x). The ECC is a cyclic code with a generator polynomial g(x) of degree n-k, and the data is encoded by dividing it by the generator polynomial. The result is a quotient q(x) of degree n-(n-k)-1 = k-1 and a remainder r(x) of at most degree n-k-1. The coefficients of the remainder r(x) are included in the code word as the n-k ECC symbols. The code word data is thus:

$$d(x)*x^{n-k} = q(x)*g(x) + r(x) \qquad \text{eqn. 1}$$

where x^{n-k} represents a shifting of the data to the appropriate position in the code word.

For a generator polynomial with two factors:

$$g(x) = g_1(x) * g_2(x)$$

where $g_1(x)$ has degree s and $g_2(x)$ has degree n-k-s, the data may be encoded by first dividing the data by the factor $g_1(x)$ and then further dividing the result by the factor $g_2(x)$. When dividing by $g_1(x)$, the data is shifted by x^s, that is, by the degree of $g_1(x)$:

$$d(x) * x^s = q_1(x) * g_1(x) + r_1(x) \qquad \text{eqn. 2}$$

where the quotient $q_1(x)$ has degree k-1+s-s = k-1 and the remainder $r_1(x)$ has at most degree s-k-1. The quotient $q_1(x)$ is then divided by the remaining factor $g_2(x)$ to produce:

$$q_1(x) = q_2(x) * g_2 + r_2(x) \qquad \text{eqn. 3}$$

where the degree of $q_2(x)$ is k-1-(n-k-s) = 2k-1-n+s, that is, the degree of $q_1(x)$ minus the degree of $g_2(x)$. Substituting eqn. 3 into the eqn. 2:

$$d(x) * x^s = (q_2(x) * g_2 + r_2(x)) * g_1(x) + r_1(x)$$

$$= q_2(x) * g_2 * g_1 + r_2(x) * g_1(x) + r_1(x). \qquad \text{eqn. 4}$$

Multiplying both sides of eqn. 4 by x^{n-k-s}, the code word data is:

$$d(x) * x^s * x^{n-k-s} = d(x) * x^{n-k} = (q_2(x) * g_2 * g_1 + r_2(x) * g_1(x)) * x^{n-k-s} + r_1(x) * x^{n-k-s}$$

$$= q_2(x) * g_2 * g_1 * x^{n-k-s} + r_2(x) * g_1(x) * x^{n-k-s} + r_1(x)^{n-k-s} \qquad \text{eqn. 5}$$

where the degree of $q_2(x) * x^{n-k-s}$ is 2k-1-n+s+n-k-2 = k-1. According to the uniqueness property of the Euclidean division theorem, the remainder of eqn. 1 equals the remainder of eqn. 5:

$$r(x) = r_2(x) * g_1(x) * x^{n-k-s} + r_1(x) * x^{n-k-s} \qquad \text{eqn. 6}$$

Multiplying by x^{n-k-s} corresponds to shifting, and thus, the remainder r(x) can be calculated directly from the remainders $r_1(x)$ and $r_2(x)$ associated with the two factors of

6

the generator polynomial g(x). Further, as discussed below, a user selects different numbers of ECC symbols by selecting to use for encoding a polynomial that includes either or both of the factors $g_1(x)$ and $g_2(x)$. The user can thus select between three polynomials, namely:

5        $p_1(x) = g_1(x)$;

       $p_2(x) = g_2(x)$ and

       $p_3(x) = g_1(x) * g_2(x) = g(x)$.

with the polynomial $p_1(x)$ producing s ECC symbols, the polynomial $p_2(x)$ producing (n-k)-s ECC symbols and the polynomial $p_3(x)$ producing n-k ECC symbols.

10   If the generator polynomial g(x) includes additional factors, the user may select various of the factors or combinations thereof as the polynomial $p_s(x)$. In addition, as discussed below with reference to Fig. 2, the factor $g_2(x)$ may itself include a number of factors, and the user may selectively use certain of the factors of $g_2(x)$ to further vary the numbers of ECC symbols included in a code word.

15   As discussed in more detail below, the code words produced in accordance with the selected polynomials $p_t(x)$ can be decoded in a conventional manner.

       B. A multi-stage encoder

       Referring now to Fig. 1, an encoder 10 for encoding data in accordance with one or both factors of a generator polynomial $g(x) = g_1(x) * g_2(x)$ includes a first stage 12 that

20 selectively divides and multiplies by the factor $g_1(x)=b_{s-1}x^{s-1}+b_{s-2}x^{s-2}+...+b_1x+b_0$ and a second stage 30 that divides the symbols supplied thereto by the factor $g_2(x)c_jx^{j-1}+...+c_1x+c_0$, where j = n-k-s. The encoder may be used to encode the data in accordance with polynomials $p_t(x)$ that include both or either of the factors $g_1(x)$ and $g_2(x)$.

25   When the polynomial $p(x) = g(x) = g_1(x)*g_2(x)$ is used for encoding, the data symbols are supplied to the first stage 12 over line 14. An adder 20 adds the contents of a

first register 18 to the data symbol and supplies the sum over line 22 as part of the quotient $q_1(x)$. With first and second switches, or multiplexers, 16 and 23 in position 1, the adder 20 provides the sum also to Galois Field multipliers 24, which multiply the sum by the coefficients of $g_1(x)$. As shown in the drawing, the first stage includes s-1 Galois Field multipliers, since the coefficient of the largest degree term of $g_1(x)$ is always a one, and thus, the associated multiplier may be omitted.

The respective products are then added in adders 22 to the contents of the associated registers 18, and the registers are updated accordingly. As each data symbol is similarly encoded, a next symbol of the quotient $q_1(x)$ is produced and the registers 18 are further updated. After the last data symbol is encoded, the switch 16 is moved to position 2, and the symbols of the remainder $r_1(x)$ are read out of the registers 18 either in parallel, or serially over line 25. The switches 16 and 23 operate under the control of a controller 27.

As the symbols of the quotient $q_1(x)$ are produced, they are supplied to the second stage 30 of the encoder. The second stage operates in a similar manner to the first stage, to divide the quotient by $g_2(x)$ and produce the remainder $r_2(x)$. The second stage produces the remainder $r_2(x)$ at essentially the same time as the first stage produces the remainder $r_1(x)$.

For the next encoding step, the remainder $r_1(x)$ is held in a processor 40 and the remainder $r_2(x)$ is fed back to the first stage 12 over line 32. The symbols of $r_2(x)$ are supplied to both an adder 28 and, through switch 23, to the Galois Field multipliers 24. The adder 28 adds the symbols of $r_2(x)$ to the contents of the first register 18 and supplies the respective sums to the processor 40 as the symbols of the product $r_2(x) * g_1(x)$. At the same time the Galois Field multipliers multiply the respective remainder symbols by the associated coefficients of $g_1(x)$ and the adders 22 add the products to the contents of the associated registers 18, to update the registers. After all of the symbols of $r_2(x)$ have been supplied to the first stage, the product $r_2(x)*g_1(x)$ is held by the processor 40.

The processor 40 next adds the product $r_2(x)*g_1(x)$ to the remainder $r_1(x)$ and multiplies, that is, shifts, the sum by x^{n-k-s} to produce n-k ECC symbols. The processor

then concatenates the ECC symbols with the data symbols that have been similarly shifted by x^{n-k-s}, to produce the n-symbol code word. The processor 40 may be replaced by appropriate shift registers and an adder (not shown).

In a second mode, the controller 27 operates the multi-stage encoder 10 to produce s ECC symbols by encoding the data in accordance with only the factor $g_1(x)$. The controller thus essentially by-passes the second stage 30 and the processor 40 includes the remainder $r_1(x)$ in the code word directly as the s ECC symbols.

Alternatively, the controller 27 may operate in a third mode to control the encoder 10 to encode the data in accordance with only the factor $g_2(x)$. The data symbols are thus supplied to the first stage 12 with the switch 16 in position 2 and the switch 23 in the position 1, such that the data symbols are supplied as $q_1(x)$ to the second stage 30. The controller 27 may instead by-pass the first stage entirely, and provide the data symbols directly to the second stage. The second stage then encodes the data symbols in accordance with $g_2(x)$ to produce the remainder $r_2(x)$. The remainder $r_2(x)$ is supplied to the processor 40, which concatenates the remainder to the data as n-k-s ECC symbols.

The encoder 10 may be used to encode data in accordance with a generator polynomial that includes more than two factors by replacing the second stage 30 with first and second sub-stages, which operate in the same manner as the stages 12 and 30 to encode the quotient $q_1(x)$ in accordance with two factors $g_3(x)$ and $g_4(x)$, where $g_2(x) = g_3(x) * g_4(x)$, and so forth.

Referring now to Fig. 2, to decode the code word produced by the encoder 10 of Fig. 1, the system encodes the code word information symbols, that is, the data, using the same polynomial that was used to produce the code word ECC symbols. The system thus uses the first and/or second stages to encode the code word data symbols by $g_1(x)$ and/or $g_2(x)$. The system next, in an error correction processor 46, compares the newly-generated ECC symbols with the ECC symbols included in the code word. If the results indicate that there are errors in the data, the processor 46 operates in a conventional manner to produce error syndromes and, if possible, correct the errors.

9

Referring now to Fig. 3, the second stage 30 of the encoder 10 may be replaced with a modified Fettweis-Hassner encoder/syndrome computation circuit 300. The Fettweis-Hassner circuit includes n-k-s, or j, Galois Field multipliers 324 that multiply the contents of associated registers 318 by the coefficients of the j-1 degree-one factors of

5    $g_2(x) = (x+\alpha^L)(x+\alpha^{L+1})...(x+ \alpha^{L+j-1})$. As in a conventional Fettweis-Hassner circuit, a switch or multiplexer 316 is controlled to supply symbols to Galois Field adders 320. In each stage, the adder 320 adds a next symbol to the results of the encoding of a previous symbol and supplies the sum to an associated register 318. The associated Galois Field multiplier 324 then multiplies the contents of the register by α^i and, in all stages except

10   the first, the Galois Field multiplier supplies the product to an associated feedback adder 326.

The feedback adder 326 adds the product to the value produced in a prior stage of the encoder and supplies the result to a next stage via an associated AND gate 328. In the first stage, the product produced by the Galois Field multiplier 324 is fed back directly to

15   the associated adder 320, and also fed through an associated AND gate 328 to the feedback adder 326 in the second stage. The AND gates link the feedback adders into a chain, and the encoding operation thus includes a latency that is associated with the delay through the chain of adders.

A multiplexer 325 is used to select which stage of the circuit 300 will operate as

20   the last stage of the encoder. For example, the multiplexer may be used to feedback to the multiplexer 316 the results produced by stage j-3, and the circuit 300 encodes by the polynomial

$$g_{2i} = (x+\alpha^L)(x+\alpha^{L+1}x)...(x-\alpha^{L+j-3})$$

to produce a remainder $r_{2i}(x)$ with j-2 symbols instead of j symbols. The multiplexer 325

25   may be used to select various other stages and the circuit then produces remainders with various numbers of ECC symbols.

By using the modified Fettweis-Hassner circuit as the second stage of the multi-stage encoder 10, a user can readily encode the data to produce a selected number of ECC symbols associated with $g_1(x)$ and/or one or more factors of $g_2(x)$. Accordingly, the

30   multi-stage encoder with the second stage 300 provides greater flexibility, without increasing the complexity of the encoder. As discussed, the factors $g_1(x)$ and $g_2(x)$ can be

selected such that the first, and faster, stage of the encoder produces a relatively large number of ECC symbols, and the delay associated with the second stage 300 is minimized by a relatively short chain of feedback adders.

Referring now to Fig. 4, to decode the code word produced using the encoder of Fig. 3 the system again encodes the data in the same manner that the system used to produce the code word ECC symbols. Thus, the system supplies the codeword data symbols to the encoder and produces ECC symbols using the first and/or second stages 12 and 300. An error correction processor 48 then compares the code word ECC symbols and the generated ECC symbols. Thereafter, as necessary, the processor produces the error syndromes associated with $g_1(x)$ in the manner discussed above with reference to Fig. 1.

To produce the error syndromes associated with the selected factors of $g_2(x)$, the system uses the circuit 300. The controller 27 thus controls the AND gates 328 to break the chain of feedback adders 326, such that each stage of the circuit 300 that was used during encoding operates independently to produce an associated error syndrome in a conventional manner. If j-3 stages are used to encode the data, the same stages are also used to produce the syndromes. This can be accomplished by selectively operating the j-3 stages or by operating all of the stages and selecting for further processing only the syndromes produced by the j-3 stages used during encoding.

The error syndromes associated with $g_2(x)$ are then provided to the processor 48, which uses the error syndromes along with those associated with $g_1(x)$ in a conventional manner, to determine error locations in the data and, if possible, correct the errors and produce error-free data.

Certain processors and controllers are shown as separate devices but may be combined. Various switches may be multiplexers or other switching devices. Further, various operations and/or steps thereof may be performed in hardware, software or firmware.

What is claimed is:

11